Building Models of Learning and Expertise with CHREST

Peter C. R. Lane (peter.lane@bcs.org.uk)
School of Computer Science, University of Hertfordshire,
College Lane, Hatfield AL10 9AB, UK

Fernand Gobet (fernand.gobet@brunel.ac.uk)
Department of Psychology, Brunel University
Cleveland Road, Uxbridge, Middlesex, UB8 3PH, UK

Keywords: CHREST; EPAM; MOSAIC; cognitive modelling; chunking; expertise; learning

Developing detailed process models of cognitive phenomena is important to the development of cognitive science as only then can cognitive theories be used to generate quantitative predictions for complex phenomena. The history of computational modelling includes many diverse approaches, from models of single phenomena (such as Sternberg's model of STM; Sternberg, 1966), to integrated models covering a wide range of different phenomena (such as Soar (Newell, 1990) and ACT-R (Anderson, 1998)), to over-arching principles, which guide the development of models in disparate domains (such as connectionist approaches, or embodied cognition).

The CHREST system, which forms the heart of this tutorial, was developed from the earlier EPAM model. EPAM provided the impetus to develop the chunking theory, which has been an important component of theories of human cognition ever since. Focussing on learning phenomena, CHREST places a great emphasis on how the model's information is learnt through interactions with an external environment. Thus, CHREST models are typically developed from large quantities of naturalistic input. For example, in modelling expert perception of chess players, actual chess games are used. Similarly, in modelling the acquisition of syntax, large corpora of mother-child interactions are employed to develop the model's long-term memory.

Objectives and Scope

CHREST (Chunk Hierarchy and REtrieval STructures) is a complete computational architecture implementing processes of learning and perception. CHREST models have successfully simulated human data in a variety of domains, such as the acquisition of syntactic categories, expertise in programming and in chess, concept formation, implicit learning, and the acquisition of multiple representations in physics for problem solving. In this tutorial, we describe the learning, perception and attention mechanisms within CHREST as well as key empirical data captured by CHREST models. Apart from the theoretical material, this tutorial also introduces participants to an implementation of CHREST and its use in a variety of domains. Material and examples are provided so participants can adapt and extend the CHREST architecture.

We begin by providing an introduction to CHREST. As an architecture, CHREST has three basic modules:

- Input/output module, which is responsible for feature extraction, passing features to the long-term memory for sorting, and guiding eye movements;
- Long-term memory, which holds information in the form a discrimination network; and
- Short-term memories, which hold pointers to nodes in the long-term memory.

Historically, CHREST is derived from the EPAM (Elementary Perceiver and Memorizer) model of Feigenbaum and Simon (1984). In both models, learning occurs through the creation and elaboration of a discrimination network. In addition, CHREST has mechanisms for the automatic construction of templates (a form of slotted schema) and for learning 'lateral links', which can be used to create elementary productions or semantic links. CHREST can thus be situated between production systems such as Soar and connectionist systems. Just as EPAM was the computational embodiment of the key aspects of the chunking theory (Chase & Simon, 1973), CHREST implements the essential aspects of the template theory (Gobet & Simon, 2000).

All the mechanisms within CHREST will be covered, including relevant aspects of the earlier EPAM model. These mechanisms include:

1. constructing a discrimination network
2. formation of templates
3. perceptual and attention mechanisms
4. the short-term memory

We will illustrate the mechanisms, individually and in combination, using domains including:

1. categorisation
2. verbal and language learning
3. implicit learning
4. chess expertise

One element which distinguishes modelling with CHREST from modelling in ACT-R or Soar is the emphasis on learning. CHREST's long-term memory is indexed through a discrimination network which must be constructed from experience in the domain of interest. For domains with high levels of human expertise, these networks become large: in the chess domain, networks comprise 100,000 to 300,000 nodes to simulate the highest levels of performance. Similar figures hold in language learning. In other domains, where experimental participants learn a task within a single session or over several days, the networks will be smaller, perhaps no more than a few hundred nodes, but even here, the networks are constructed by CHREST using its learning mechanisms. Thus, unlike ACT-R or Soar, CHREST is not 'programmed' with information about a task, but is presented with data (such as chess positions), and its behaviour is then observed. Indeed, this technique makes CHREST more akin to standard practice in connectionist modelling, where networks are trained. However CHREST operates at the level of meaningful symbols.

CHREST Software

The latest CHREST implementation will be demonstrated and described. CHREST is implemented in Java and comes complete with a flexible graphical interface. The graphical environment enables users to build CHREST models from data provided within an input data file. The implementation also supports the construction of more complex or novel experimental setups through a programmable interface: any Java-aware language may be used to develop models, and several complete examples are provided with the distribution.

Within the tutorial we will introduce participants to the graphical environment, walk them through a number of provided examples which will illustrate the workings of the architecture and some samples of successful applications, and finally describe the input data format for applying the environment to new domains. Documentation and examples are available for participants wanting to look at the code and use it as a library for writing more complex models.

Target Audience

The tutorial is designed to introduce the area of symbolic computational modelling using the chunking/template theories, as implemented in CHREST. We have designed the material to be suitable for newcomers to the fields of either cognitive science or cognitive modelling. We assume some understanding of human psychology and learning, plus experimental methods, appropriate to graduate students in this area. As the domains covered by CHREST are not frequently tackled by other symbolic cognitive models, we would expect the level to also be appropriate for more advanced researchers who have not previously studied EPAM or CHREST.

Participants are recommended to bring a laptop with the software installed (only Java 6 is required), to follow along the demonstration parts of the tutorial.

Further Materials

Participants receive a CD-ROM containing copies of the software, lecture slides and supporting papers. The latest version of the software and further links are available from `http://chrest.info`

Schedule of Events

1st half (80 minutes)	
Introduction to tutorial	5 min
Overview of CHREST	20 min
Learning mechanisms in CHREST	15 min
Demonstration of CHREST shell and learning processes, by tracing a working model	15 min
Key experimental results (e.g. verbal-learning phenomena, MOSAIC, EPAM-VOC)	25 min
Coffee break	20 min
2nd half (80 minutes)	
Key experimental data about expertise	15 min
Perceptual mechanisms in CHREST	15 min
Demonstration of CHREST models of perceptual expertise, using key experiments in chess.	20 min
Description of input format, so participants can develop CHREST models for their own domains.	15 min
Questions	10 min
Conclusion	5 min

References

Anderson, J. R., & Lebière, C. (Eds.). (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, *4*, 55-81.

Feigenbaum, E. A., & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science*, *8*, 305–336.

Freudenthal, D., Pine, J. M., Aguado-Orea, J., & Gobet, F. (2007). Modelling the developmental patterning of finiteness marking in English, Dutch, German and Spanish using MOSAIC. *Cognitive Science*, *31*, 311–341.

Gobet, F., Lane, P. C. R., Croker, S. J., Cheng, P. C.-H., Jones, G., Oliver, I., et al. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, *5*, 236–243.

Gobet, F., & Simon, H. A. (2000). Five seconds or sixty? Presentation time in expert memory. *Cognitive Science*, *24*, 651–82.

Jones, G. A., Gobet, F., & Pine, J. M. (2007). Linking working memory and long-term memory: A computational model of the learning of new words. *Developmental Science*, *10*, 853–873.

Lane, P. C. R., Gobet, F., & Ll. Smith, R. (2009). Attention mechanisms in the CHREST cognitive architecture. In L. Paletta & J. K. Tsotsos (Eds.), *Proceedings of the fifth international workshop on attention in cognitive science* (Vol. LNAI 5395, pp. 183–196). Berlin: Springer-Verlag, GmbH.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Sternberg, S. (1966). High speed scanning in human memory. *Science*, *153*, 652–654.